

DailyLeadProcessor.apxc

```
public without sharing class DailyLeadProcessor implements Schedulable {

    public void execute(SchedulableContext ctx) {

        List<Lead> leads = [SELECT ID, LeadSource FROM Lead WHERE LeadSource = null
LIMIT 200];
        for ( Lead l : leads) {
            l.LeadSource = 'Dreamforce';
        }

        update leads;
    }
}
```

DailyLeadProcessorTest.apxc

```
@isTest
public class DailyLeadProcessorTest {

    private static String CRON_EXP = '0 0 0 ? * * *';

    @isTest
    private static void testSchedulableClass(){

        List<Lead> leads = new List<lead>();
        for (Integer i=0; i<500; i++){
            if ( i < 250) {
                leads.add(new Lead(LastName='Connock', Company='Salesforce'));
            } else {
                leads.add(new Lead(LastName='Connock', company='Salesforce',
LeadSource= 'Other'));
            }
        }
        insert leads;

        Test.startTest();
        string jobId = System.schedule('process Leads', CRON_EXP, new
DailyLeadProcessor());
        Test.stopTest();

        List<Lead> updatedLeads = [SELECT ID, LeadSource FROM Lead WHERE Leadsouce
= 'Dreamforce'];
        System.assertEquals(200, updatedLeads.size(), 'ERROR: At least 1 record not
updated correctly');

        List<CronTrigger> cts = [SELECT ID, TimesTriggered, NextFireTime FROM
```

```
CronTrigger WHERE ID = :jobId];  
    System.debug('Next Fire Time' + cts[0].NextFireTime);  
    }  
}
```